Applied Mathematical Modelling 36 (2012) 6301–6311

Contents lists available at SciVerse ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Adjacency based method for generating maximal efficient faces in multiobjective linear programming

S. Krichen [a,b,*], H. Masri [b], A. Guitouni [c,d]

[a] Faculty of Law, Economics and Management, University of Jendouba, Avenue de l'UMA, 8189 Jendouba, Tunisia
[b] LARODEC Laboratory, Institut Supérieur de Gestion de Tunis, University of Tunis, Rue de la liberté, 2000 Bardo, Tunisia
[c] Command and Control Decision Support System Section, Defence Research and Development Canada (DRDC) Valcartier, 2459 Pie-XI North, Quebec (Quebec), Canada G3J 1X5
[d] Faculty of Business, University of Victoria, Victoria, British-Columbia, Canada

ARTICLE INFO

Article history:
Received 15 July 2010
Received in revised form 6 January 2012
Accepted 17 January 2012
Available online 9 February 2012

Keywords:
Multiobjective linear programming
Adjacency
Efficient set
Efficient faces
Outcome set

ABSTRACT

Multiobjective linear optimization problems (MOLPs) arise when several linear objective functions have to be optimized over a convex polyhedron. In this paper, we propose a new method for generating the entire efficient set for MOLPs in the outcome space. This method is based on the concept of adjacencies between efficient extreme points. It uses a local exploration approach to generate simultaneously efficient extreme points and maximal efficient faces. We therefore define an efficient face as the combination of adjacent efficient extreme points that define its border. We propose to use an iterative simplex pivoting algorithm to find adjacent efficient extreme points. Concurrently, maximal efficient faces are generated by testing relative interior points. The proposed method is constructive such that each extreme point, while searching for incident faces, can transmit some local informations to its adjacent efficient extreme points in order to complete the faces' construction. The performance of our method is reported and the computational results based on randomly generated MOLPs are discussed.

1. Introduction

Many real optimization problems can be modeled as a multiobjective linear programming (MOLP) problem, such as transportation [1] and inventory planning [2]. MOLPs are characterized by the simultaneous maximization of a set of objectives under a system of constraints, where both objective functions and constraints are linear, within continuous decision variables. Solving MOLPs yields to a set of efficient solutions belonging to a connected set of efficient faces. The set of all efficient solutions is the union of all maximal efficient faces. Several approaches for solving such problems have been proposed in the literature [3–5]. These methods employ different schemes for exploring the efficient set, and different ways of characterizing and locating the maximal efficient faces.

Recently, much more attention has been given to solving MOLPs in objective space. The earlier works of Dauer and Gallagher [6] and Benson [7] provided a new view for studying efficient solutions set in outcome space. Comparing to the decision based approaches, the outcome based search methods seem to give promissive computational results [8]. Yan et al. [9] presented a method for generating maximal efficient faces in combined decision-outcome space using the weight decomposition and with an easy and clear solution structure, they defined the set of efficient faces as a finite combination of

* Corresponding author at: Faculty of Law, Economics and Management, University of Jendouba, Avenue de l'UMA, 8189 Jendouba, Tunisia
E-mail addresses: saoussen.krichen@isg.rnu.tn (S. Krichen), masri_hela@yahoo.fr (H. Masri), Adel.Guitouni@drdc-rddc.gc.ca (A. Guitouni).

0307-904X/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.apm.2012.01.034

extreme points and rays. This representation was called the sum form. More details might be found is Section 3. Most of exiting outcome based methods, operate in two steps: generating efficient extreme points then locating efficient faces.

In this paper, we propose a new method for solving MOLPs in objective space, by combining the sum form approach with the adjacency information of efficient extreme points. The proposed method performs the neighborhood approach defined by Benson and Sun [10] in order to generate efficient extreme points. Simultaneously, the efficiency of certain combinations of these extreme points are tested by investigating relative interior points in order to find maximal efficient faces. The main idea is to design a local and constructive method. We associate to each extreme point x a list $L(x)$ to keep in memory the faces containing this point. At each iteration, we investigate a new extreme point. We start by exploring its adjacent efficient vertices. If a new efficient point x' is generated, then we test the efficiency of the convex hull defined by combining this new point x' with preexisting faces in $L(x)$. To do so, a relative interior point is tested. If a new face is encountered, we update the list $L(x')$ by adding this face. This procedure iterates until exploring all efficient extreme points.

The paper is structured as follows. In the next section, we present the multiobjective linear programming problem and some theoretical background. In Section 3, we review related literature. In Section 4, we present the proposed method for generating maximal efficient faces in outcome space. We provide empirical results in Section 5. A comparison with an existing method [5], as well as a computational analysis for medium and large sized problems are also discussed in the same section. The conclusion is presented in Section 6.

2. Problem description and fundamental results

A MOLP problem can be defined as:

$$\begin{aligned} \text{Min} \quad & Cx \\ \text{s.t} \quad & Ax \leqslant b, \\ & x \geqslant 0, \end{aligned} \tag{1}$$

where C is a $p \times n$ objective functions matrix. A is an $m \times n$ matrix of constraints coefficients, $b \in R^m$, and $x \in R^n$ is the vector of decision variables.

Let $X = \{x \in R^n | Ax \leqslant b\ x \geqslant 0\}$ be the feasible set in the decision space which is defined by the set of constraints. The feasible set in the objective space is $Y = CX = \{Cx | x \in X\}$. Y is also called the outcome set.

In multiobjective context, usually there is no unique optimal solution, but rather a set of efficient or non dominated solutions. So, optimality is replaced by efficiency.

Definition 1. A solution x_0 dominates another solution x_1 if and only if $Cx_0 \leqslant Cx_1$ and $Cx_0 \neq Cx_1$ i.e x_0 is no worse than x_1 according to all objectives and x_0 is strictly better than x_1 in at least one objective (minimization problem) [11].

Definition 2. A solution x_0 is said to be pareto optimal or efficient if and only if $\nexists x_1 \in X : x_1$ dominates x_0 [11].

In objective space, an efficient solution is defined similarly:

Definition 3. A solution $y_0 \in Y$ is called an efficient outcome solution if and only if $\nexists y_1 \in Y$: $y_1 \leqslant y$ and $y_1 \neq y$ [12].

Therefore, the pareto optimal set in decision space is the set of all efficient solutions X_e. And the efficient outcome set Y_e is exactly the mapping of X_e using the objectives matrix C, such that $Y_e = \{Cx | x \in X_e\}$.

In this paper, we suppose that X is a nonempty compact polyhedron. Hence, it can be easily proven that Y is also a nonempty compact polyhedron [6]. Let X_{ex} the set of all efficient vertices in decision space. Each extreme point $x \in X_{ex}$ can be mapped by C either into a single extreme point or into an non extreme point in the outcome set Y. However, for each extreme point $y \in Y_{ex}$ there exists at least one element $x = Cy$ such that $x \in X_{ex}$ [10]. One of the most important results, is that the set X_{ex} (or Y_{ex}) is connected [11]. The connectedness means that for every two efficient points, there is necessarily an efficient path between them (efficient edges relating them). This result makes a method such that simplex convenient and adaptable for exploring the efficient solutions.

Definition 4. A face is located in the boundary of the polyhedron X (resp Y). It is the intersection of X (Y) with a supporting hyperplane. A face is characterized by its dimensionality: vertices are of zero dimension and edges are one dimension, while the maximal dimension is $n-1$ ($p-1$) [3].

A face F is an efficient face if and only if every point on F is efficient [7]. Let $F \subset X_e$ (resp $F \subset Y_e$) be an efficient face of X (Y), F is called maximal efficient face, if there is no efficient face F_1 of higher dimension such that $F \subset F_1$ [3]. Hence, solving a multiobjective linear problem can be reduced to finding all maximal efficient faces.

The efficiency of a face can be investigated by testing a relative interior point. Thus, let F be an arbitrary convex subset of X (resp Y), and let $x \in$ the relative interior of F. If x is efficient than all the face is efficient and if x is dominated then all the relative interior set is dominated [4].

This result implies that an efficient face can be defined as the convex hull of the efficient extreme points belonging to this face. Also, a face can be defined by a weight vector such that the solution of the weighted-sum problem is all the points of this face. These two representations are called the sum form and the weight form.

3. Related literature

The earlier methods that attempted to solve MOLPs were based on the simplex method [13]. In fact, the connectedness of the efficient set enabled the use of such approach. These algorithms employed various search schemes to iteratively identify and test the efficiency of faces. They can be classified according to their resolution approach, into three categories: bottom-up, top-down and mixed approaches. Bottom up methods [3,14–16] generate iteratively efficient extreme points and maximal faces incident to these points. Besides, mixed approaches [4,17,18] consist on decomposing the problem in two steps: generating efficient extreme points, then identifying maximal efficient faces. More recently, Yan et al. [9] presented a combined decision-objective space method for generating maximal efficient faces using a finite number of weights, that enables finding efficient points. They present an easy and clear solution structure, by defining the set of efficient faces as a finite combination of extreme points and rays (sum form). This work was later slightly modified by Foroughi and Jafari [19] to overcome the problem of unbounded weighted sum problems. Poukarimia et al. [18] proposed a similar approach by exploiting the affine independent property between efficient extreme points. For the top down approach, the method of Sayin [5] is the only method figuring in this class. Its idea is strongly inspired from [4], because it uses the facial decomposition. It disregards the step of finding efficient extreme points, and proceeds directly by determining efficient faces of higher dimension. The dimension is then decreased iteratively.

Recently, several research attempted to solve MOLPs by generating the efficient outcome set [9,19,18]. The principal advantage is that generally the number of dimensions of the outcome space is significantly smaller than the decision space, since the number of objective functions is less than the number of decision variables, and due to collapsing that occurs while mapping X to Y. All methods operating in decision space face severe drawbacks. First, the computational demands of finding all efficient set grow rapidly with problem size [20], so that only small size problems could be solved in the decision space. Second, size and nature of the efficient decision set generally lead to difficulties in communicating it to the decision maker [8]. Finally, it is known that the decision maker prefers to analyze the efficient solution tradeoffs in the objective space rather than in the decision space.

The works of [7,6] constitute the basis for most outcome set based methods. They defined the sufficient conditions to characterize efficient extreme points and efficient faces in objective space, as well as the correspondence between the efficient set in decision space and in outcome space. Consequently, several methods were proposed based on the search in the outcome space. Most of these methods deal only with generating the set of efficient extreme points. Different procedures were developed. Benson and Sun [21,22] investigated the decomposition of the weight space to generate extreme points. The same author also proposed an adaptation of an outer approximation algorithm to find the set of efficient extreme points. Few research studied the generation of maximal efficient faces in objective space. Benson [22] investigated the bi-criteria case.

4. Efficient solutions adjacency based method (ESAM)

The proposed method is motivated by the following ideas:

- A face is described as the convex hull of all extreme points defining it. If a face is efficient than all the points belonging to the relative interior and the closure of this face are necessarily efficient.
- The efficiency of a face can be verified by testing the efficiency of a relative interior point belonging to this face.

Considering these theoretical results, the main idea of the proposed method is to exploit the adjacency information among efficient extreme points while using a simplex pivoting logic. Therefore, ESAM explores the efficient extreme points, constructs and tests iteratively some combinations of these extreme points that define the frontier of efficient faces. Before presenting the algorithm, we should introduce the following notations.

Notation:

- y: The vector defining the coordinate of an efficient extreme point,
- (y,y'): The edge (y,y'),
- $L(y)$: The list of efficient faces incident to the point y,
- L_{exp}: The list of explored efficient extreme points,
- L_{ex}: The list of generated efficient extreme points,
- L_F: The list of the maximal efficient faces.

In what follows, we detail the different steps of the algorithm. The algorithm is composed of two main steps. In the initialization step, a first efficient point is generated. Then, the second step explores the efficient extreme points and generates iteratively all maximal efficient faces.

The general scheme of the algorithm is as follows:

The adjacency based algorithm

Initialization
  Generate a first efficient extreme point
Iterative process
  Given a current unexplored efficient extreme point y
  Update L_{exp}
  test-combination $(L(y))$ (to test the combination of the faces contained in $L(y)$ and update $L(y)$ and L_F)
  for all adjacent efficient extreme points $y' \notin L_{exp}$ **do**
    Update L_{ex}
    $L(y') = L(y') \bigcup (y,y')$
    $L_F = L_F \bigcup (y,y')$
    for all $F \in L(y)$ **do**
      if the face $Conv(y' \bigcup F)$ is efficient **then**
        $L(y') = L(y') \bigcup (y' \bigcup F)$ and delete $subfaces \in L(y')$
        $L_F = L_F \bigcup (y' \bigcup F)$ and delete $subfaces \in L_F$
      end if
    end for
  end for
Stop All efficient extreme points were explored
Output All maximal efficient faces L_F
Procedure-test-combination$(L(y))$
  $size$ = size of $(L(y))$
  for $i = 1$ to $size - 1$ **do**
    $List = \emptyset$, $j = i + 1$;
    while $j<$ size of $L(y)$ **then**
      test the efficiency of $(L(y)[i] \bigcup L(y)[j])$
      if efficient **then**
        $list = list \bigcup (L(y)[i] \bigcup L(y)[j])$
        Update L_F
      end if
    end while
    $L(y) = L(y) \bigcup list$
  end for

① Initialization: In order to generate a first efficient extreme point, ESAM uses Benson's method [10], which is strongly inspired from the sequential maximization [13]. This procedure constructs a sequence of nested subsets in the feasible region. The process begins by optimizing the first objective. Then, the obtained optimal solution y_1 defines the feasible region for a second program that optimize the second objective, and so on and so forth until finding a unique optimal solution. This solution will correspond to the first efficient extreme point in the outcome space. As a result of this step, one can deduce that the problem is infeasible, or there is at least one efficient extreme point.

② Iterative process: Generate all efficient extreme points and maximal efficient faces in an iterative way. For each extreme point y, we define a list $L(y)$ containing all the generated efficient faces that contain this point. The process starts from an unexplored efficient point, if the list $L(y)$ contains more than one face then the method tests the efficiency of the combination of these faces. This procedure aims at finding new faces by combining the elements of the list $L(y)$.

Then, all adjacent vertices are investigated. For each new extreme point y', both lists $L(y')$ and the list of efficient faces L_F are updated by adding the edge (y,y'), this edge is included only if there is no higher face in $L(y')$ such that $(y,y') \subset L(y')$. Then, the convex hull defined by the union of y' and each element $F \in L(y)$ is analyzed to determine its efficiency. If new efficient faces are found, then $L(y')$ and L_F are updated. The process continues by moving to another unexplored extreme point, and so on and so forth.

The adopted pivoting rule and the linear program for testing a relative interior point are described in what follows. It is important to mention that the proposed algorithm can also be applied to the decision space, using the multiobjective simplex method to generate the efficient extreme points [13], and the same approach for finding efficient faces. Hence, the modification of the search space is feasible to the detriment of the computational requirements. The used notation should not create any confusion, we mean by y an extreme efficient point in outcome or decision space.

- **Pivoting rule:** The efficient extreme points are generated using the outcome pivoting method [10]. This method gives a neighboring procedure that enable to move from a current extreme point to all its adjacent vertices. A test of efficiency at each encountered point is done to guarantee its efficiency. To avoid the degeneracy problem, Gal an Geue's procedure is then used [23].
- **Testing a relative interior point:** A point y belonging to the relative interior of the corresponding face F is selected. y should satisfy this condition: $y = \frac{\sum_{i=1}^{k} y_i}{k}, y_i \in Y_{ex} \bigcup F$ and k the number of efficient extreme points defining the face F. Then, we solve the following linear program:

$$\begin{aligned} \text{Max} \quad & e^T v, \\ \text{s.t.} \quad & Cx - Iv = y, \\ & x \in X, \quad v \geqslant 0, \end{aligned} \tag{2}$$

with $e^T = (1, \ldots, 1) \in R^p$ and I is the identity matrix. if the value of the objective function for the optimal solution of the linear program is nil, then y will necessary be an efficient solution for the MOLP. So, the face F is also efficient because y is a relative interior point of F. This LP can be slightly modified if we are testing an extreme point in decision space.

5. Illustrative example

In order to illustrate the proposed algorithm, let us consider the example proposed in [6], where $n = 3$, $m = 8$ and $p = 2$: (See Fig. 1)

$$\begin{aligned} \text{Min} \quad & x_1 + 0.11x_2, \\ & x_2 + 0.11x_3, \\ \text{s.t.} \quad & 9x_1 + 9x_2 + 2x_3 \leqslant 81, \\ & 8x_1 + x_2 + 8x_3 \leqslant 72, \\ & x_1 + 8x_2 + 8x_3 \leqslant 72, \\ & 7x_1 + x_2 + x_3 \geqslant 9, \\ & x_1 + 7x_2 + x_3 \geqslant 9, \\ & x_1 + x_2 + 7x_3 \geqslant 9, \\ & x_1 \leqslant 8, \\ & x_2 \leqslant 8, \\ & x_1, x_2, x_3 \geqslant 0. \end{aligned} \tag{3}$$

To appreciate the advantages of exploring the outcome space rather than the solution space, we illustrate the application of the proposed algorithm in both decision and outcome space. The feasible set in decision space is shown in Fig. 1. We have 11 extreme points, 7 of them are efficient. **Initialization**: $L_{ex} = \emptyset$, $L_{exp} = \emptyset$, $L_F = \emptyset$

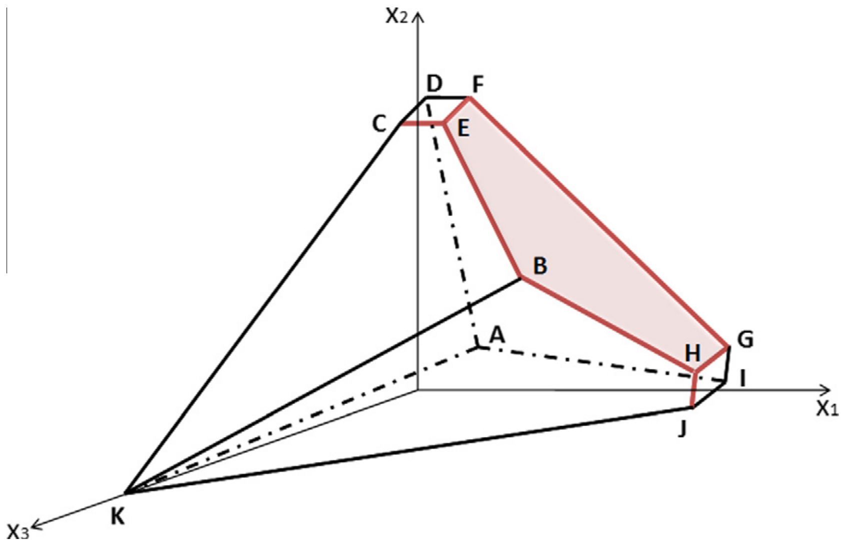

Fig. 1. Feasible set in decision space.

We suppose that the first obtained efficient point is G. $L_{ex} = \{G\}$.
Iterative process: In each iteration, we explore a new efficient extreme point

–**Iteration 1**: ESAM starts by exploring the point G, $L_{exp} = \{G\}$.
Using the multiobjective simplex table corresponding to this point, two adjacent efficient vertices H and F are found.

- For the first point H: The list of efficient points $L_{ex} = \{G,H\}$ is updated, then the lists of efficient faces $L(H) = \{(G,H)\}, L_F = \{(G,H)\}$ are also updated. The list $L(G)$ is empty so there is no more test to be carried out.
- For the point F: $L_{ex} = \{G,H,F\}, L(F) = \{(G,F)\}, L_F = \{(G,H),(G,F)\}$. The new generated efficient extreme points is explored in the next iterations.

–**Iteration 2**: The current point is H, $L_{exp} = \{G,H\}$. The set of adjacent efficient vertices to H contains G, B and J.

- The point $G \in L_{exp}$, so there is no further tests to do.
- For the point B: $L_{ex} = \{G,H,F,B\}, L(B) = \{(H,B)\}, L_F = \{(G,H),(G,F),(H,B)\}$.
  In this step, we notice that $L(H) \neq \emptyset$ so the efficiency of the convex hull defined by the union of the new encountered edges and the faces stored in the current list $L(H)$ is tested. Hence, a relative interior point $x \in (G,H) \bigcup (H,B)$is selected. By solving the LP (2), one can verify that x is efficient, therefore a new efficient face is found. The two lists $L(B) = \{(H,B),(H,B,G)\}$, $L_F = \{(H,B,G),(G,F)\}$ are updated.
- For the point J: $L_{ex} = \{G,H,F,B,J\}$, $L(J) = \{(H,J)\}$, $L_F = \{(H,B,G),(G,F),(H,J)\}$. The method tests the efficiency of $(H,J) \bigcup (H,B)$. This combination is not efficient.

–**Iteration 3**: The current point is F, $L_{exp} = \{G,H,F\}$. The adjacent extreme points are G, E.

- The point G is already explored.
- For the point E: $L_{ex} = \{G,H,F,B,E\}$, $L(E) = \{(E,F)\}$, $L_F = \{(H,B,G),(G,F),(H,J),(E,F)\}$.
  The efficiency of the convex hull defined by $(E,F) \bigcup (G,F)$ is tested: it is efficient. Hence, $L(E) = \{(E,F),(E,F,G)\}$, $L_F = \{(H,B,G),(H,J),(E,F,G)\}$.

–**Iteration 4**: The current point is B, $L_{exp} = \{G,H,F,B\}$. $|L(B)| > 1$, so procedure $test - combination(\,)$ is executed. As a result no efficient faces are generated and no changes are made to L_F and $L(B)$. Then, adjacent points are explored and H, E vertices are found.

- The point H is already explored.
- The second point E: $L(E) = \{(E,F),(E,F,G),(E,B)\}$.
  $(E,B) \bigcup (H,B)$ is efficient, then $L(E) = \{(E,F),(E,F,G),(E,B),(E,B,H)\}$ and $L_F = \{(H,B,G),(H,J),(E,F,G),(E,B,H)\}$.
  Also $(E,B) \bigcup (H,B,G)$ is efficient, then $L(E) = \{(E,F),(E,F,G),(E,B),(E,B,H),(E,B,H,G)\}$ and $L_F = \{(H,J),(E,F,G),(E,B,H,G)\}$.

–**Iteration 5**: The current point is J, $L_{exp} = \{G,H,F,B,J\}$. This point has just one adjacent extreme point B, which is already explored.
–**Iteration 6**: The current point is E, $L_{exp} = \{G,H,F,B,J,E\}$. $|L(E)| > 1$, therefore the procedure $test - combination(\,)$ is called. By investigating all the combination of all the faces in $L(E)$, a new face (E,B,H,G,F) is located, then $L_F = \{(H,J),(E,B,H,G,F)\}$. After that, ESAM proceeds by searching the extreme adjacent points, we found F, B and C

- B and F are explored.
- The third point C: $L(C) = \{(E,C)\}$, $L_F = \{(H,J),(E,B,H,G,F),(E,C)\}$. By testing the efficiency of (E,C) with all elements of $L(E)$, we find that no new faces exist.

–**Iteration 7**: The last point is C, $L_{exp} = \{G,H,F,B,J,E,C\}$. No new faces are encountered. There is no more efficient points to explore, hence, we conclude that the set of maximal efficient faces is $L_F = \{(H,J),(E,B,H,G,F),(E,C)\}$.

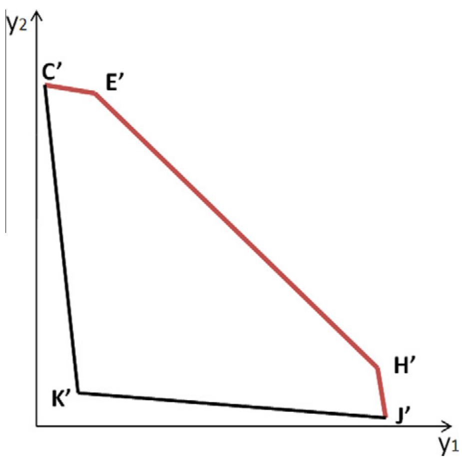

Fig. 2. Feasible set in outcome space.

For objective space, the feasible set is presented in Fig. 2. It contains 5 extreme points, and 4 efficient points.
Initialization: $L_{ex} = \emptyset, L_{exp} = \emptyset$, $L_F = \emptyset$
Let us suppose that the first obtained efficient point is C'. $L_{ex} = \{C'\}$
Iterative process:

–**Iteration 1:** (Exploring the first efficient extreme point in the objective space) $L_{exp} = \{C'\}$.
This point has only one adjacent extreme point E'. Therefore, the list of efficient points $L_{ex} = \{C', E'\}$ is updated. Then the lists of efficient faces $L(E') = \{(C', E')\}$, $L_F = \{(C', E')\}$ are also updated.
–**Iteration 2:** The current point is E', $L_{exp} = \{C', E'\}$.
E' has one adjacent extreme point H'. Hence, $L_{ex} = \{C', E', H'\}$, $L(H') = \{(E', H')\}$, $L_F = \{(C', E'), (E', H')\}$.
The convex hull defined by $(C', E') \bigcup (E', H')$ is tested. This combination is not efficient.
–**Iteration 3:** The current point is H', $L_{exp} = \{C', E', H'\}$.
H' has only one adjacent vertex J'. Hence, $L_{ex} = \{C', E', H', J'\}$, $L(J') = \{(J', H')\}$, $L_F = \{(C', E'), (E', H'), (J', H')\}$.
The efficiency of relative interior point $\in (J', H') \bigcup (E', H')$ is tested: it is not efficient.
–**Iteration 4:** Finally, the point J' is explored, $L_{exp} = \{C', E', H', J'\}$. We find only one adjacent vertex H' which is already explored. So, no more faces are generated. Hence we conclude that the list of maximal efficient faces is $L_F = \{(C', E'), (E', H'), (J', H')\}$.

We notice that the number of iterations to solve this problem in outcome space is less than the decision space. Therefore, the computational burden in objective space is generally reduced. Eventhough it has been proven that the maximal efficient faces in X and in Y are in one to one correspondence [6], the structure and the dimension of the efficient outcome set is generally easier to handle.

These observations line up with the results stated in the literature for comparing the complexity of solving MOLPs in decision and outcome space. As stated before, ESAM could be applied to both spaces, by varying the pivoting rule. We should first mention that the complexity of ESAM depends to a large extent on two following parameters:

- The number of extreme points in the considered decision or outcome space: it defines implicitly the global number of iterations of the algorithm. Due to collapsing that occurs while mapping X to Y, we can conclude that the number of iterations will be reduced while solving the problem in the outcome space.
- The neighboring of each extreme point: This value determines how much work is done in one iteration of ESAM. Basically, an iteration consists on exploring the neighboorhood of a current extreme point y and testing the efficiency of the combination of each adjacent extreme point y' and the previous encountered faces listed in $L(y)$. Here, we should notice that the maximal number of adjacent vertices in the case of non degeneracy is equal to p in outcome space and n in decision space. Generally, since the number of objective functions is significantly lower than the number of decision variables (i.e $p \ll n$), applying ESAM to the outcome space is computationally less consuming than the decision space.

6. Empirical results

We performed many computational experiments for different problem sizes in order to validate the performance of the proposed algorithm (ESAM). The algorithm is coded in C++. An open source library CoinMP is used to solve uni-objective linear problems. A Core2 duo 2*GHZ* laptop has been used for these experiments.

This experiment design is into folds:

① First, we compare the ESAM with an existing method [5] operating in the decision space. The comparison between the two methods is based on their speed to solve randomly generated multi-objective problems: generating the efficient solutions set as well as the maximal efficient faces. Few papers have conducted computational analysis [3,17,5]. We chose the Sayin's method because we kept the same empirical design defined in [5].
② Second, we propose to characterize the proposed algorithm by providing empirical results showing its performance: CPU time versus problem size. We should mention that this characterization is performed from a descriptive perspective.

To do so, many MOLPs are randomly generated for different sizes based on: the number of objectives p, the number of constraints m and the number of variables n.

6.1. Comparison between ESAM and Sayin's method

For each problem instance, 10 MOLPs have been randomly generated. Table 1 summarizes the principal results obtained by both ESAM and Sayin's methods. Y_{ex} is the average number of efficient extreme points and F is the average number of maximal efficient faces. The average CPU time for both methods is given in seconds.

Based on the results of Table 1, one can observe that the method of Sayin seems to be very depended on the number of variables n. Even small variation of n could significantly increase the CPU time. For example, the CPU time jumps from about 24 s for MOLP ($p = 3$, $m = 3$, $n = 9$) to more than 142 s for MOPL ($p = 3, m = 3, n = 10$). The proposed ESAM is less influenced by

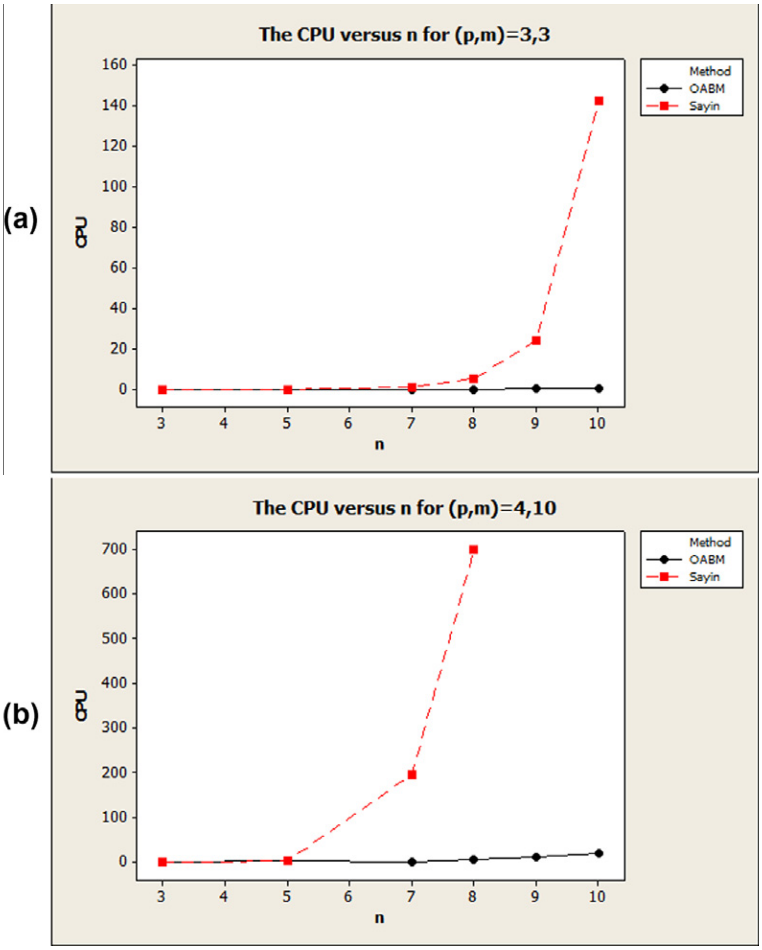

Fig. 3. Comparison of the CPU time in terms of number of variables n: (a): $(p,m) = (3,3)$ (b): $(p,m) = (4,10)$.

the number of decision variables. The difference might be explained by the search difference between exploring the decision or the outcome spaces. The ESAM appears to be more sensitive to the number of objective functions than the Sayin's method. One should also observe that Sayin's method does not converge for many small size MOLPs, while the ESAM method continues to generate the efficient solutions as well as efficient faces.

Fig. 3 shows the variation of the CPU time for both methods in function of n for a fixed value of the couple of variables p and m. For small values of n ($n < 5$), the two methods seems to have similar performance. But, when n is increased ESAM outperform Sayin's method. For example, for $n = 10$, $m = 10$, the average CPU time required by Sayin's method is greater than 700 s. However, for small n and large p ($p > 7$), Sayin's method seems to better perform than the ESAM.

6.2. Additional empirical results for ESAM

An additional set of medium size problems are randomly generated, with greater values of n and m. We have limited the number of objectives to $p = 2, 3$. Table 2 shows the obtained results. It is important to observe that the CPU time increases rapidly with the problem size in particular with the number of objectives. Indeed, the number of extreme points and the number of faces of the feasible outcome region are directly related to p (it determines the dimension of Y).

Fig. 4 is a 2-dimensional graph, representing the relation between CPU, m and n with $p = 3$. We notice that the tendency of this surface plot has an exponential form, as n increases. Fig. 5 is a plot of the number of efficient face versus n and m. We notice that the form of the curves in this figure are similar to Fig. 4. We conclude that the number of maximal efficient faces and the CPU time are positively correlated. Hence, the more the structure of Y_E is complex and large, the more the resolution of the MOLP is computationally demanding. Therefore, analyzing the CPU time or the number of efficient faces versus the other parameters will remain to the same conclusions.

This method seems to give promissive results for small and medium sized problem, with a reasonable CPU time. But, we are inherently limited by the storage requirement that increases rapidly with the problem size. As one can notice, the

Table 1
A comparison between ESAM and Sayin's method.

p	*m*	*n*	Y_{ex}	*F*	ESAM CPU	Sayin CPU
		3	4.25	2	0.036	0.019
		5	5	2.750	0.095	0.155
	3	7	6	3.25	0.107	1.235
		8	7.66	4	0.182	5.46
		9	10.75	6.25	0.283	24.105
3		10	11.75	9	0.383	142.5
		3	3	1.25	0.048	0.055
		5	10.25	4.5	0.233	1.288
	10	7	12.75	7.75	0.303	79.1
		8	13.66	7.33	0.37	653
		9	21.5	12.75	0.625	–
		10	24	15.5	0.740	–
		3	4.5	1.75	0.073	0.033
		5	15.66	5.33	1.183	1.1168
4	10	7	15.667	6.33	0.55	195
		8	30	12.33	4.233	698
		9	33.75	15.25	9.325	–
		10	37.667	15.66	18.63	–

Table 2
Computational results for ESAM ($p = 2,3$).

p	*m*	*n*	Y_{ex}	*F*	CPU
	50	50	14.5	13.5	1.765
	100	100	22.5	21.5	9.55
2	100	200	29.6	28.6	42.64
	200	100	38.33	37.33	26.45
	200	200	53.5	52.5	65.69
	3	20	10.66	6	0.43
	10	20	51.75	39	2.7
	20	3	7	3.5	0.145
3	20	10	42	26.5	1.68
	20	20	73.25	57.5	5.178
	3	60	13.5	9	0.75
	10	60	167.25	153	14.2
	20	60	385.25	348.75	41.25
	60	60	900	750	210

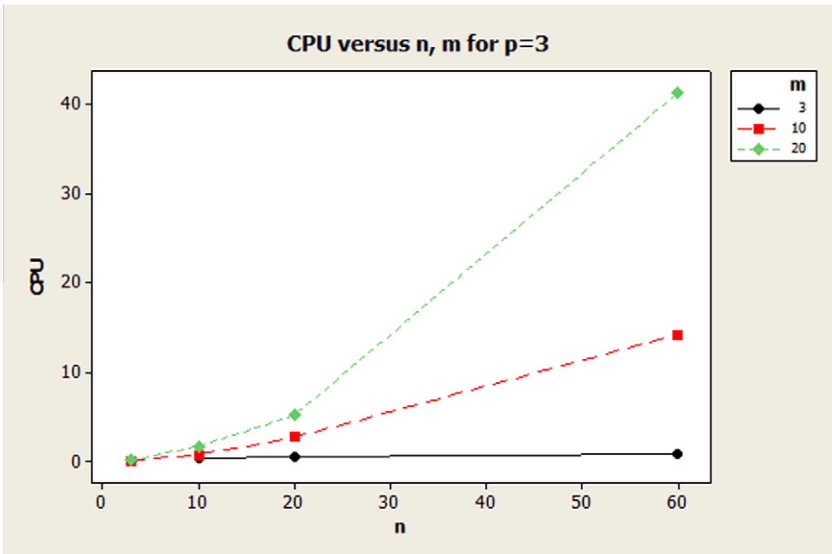

Fig. 4. CPU time in terms of *n* and *m*.

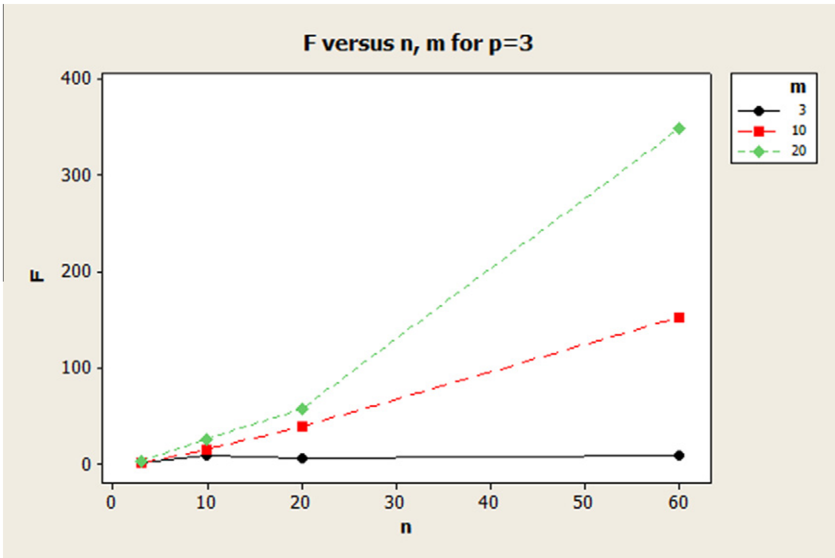

Fig. 5. Number of faces in terms of n and m.

number of extreme points is dependent of the three size parameters and specially p. While solving a MOLP problem, each point will have its own list of efficient faces $L(x)$ (all faces that contain this point and not necessary maximal), the size of this list is dependent of the dimension of the faces. Therefore, solving a MOLP with a large number of objective functions can require an important volume of memory.

7. Conclusion

This paper propose an adjacency based method for generating maximal efficient faces for solving MOLPs in the objective (outcome) space called efficient solutions adjacency based method (ESAM). ESAM is based on the simplex pivoting method [10] in order to generate efficient extreme points, and then it constructs simultaneously efficient faces by testing selected relative interior points. It exploits the properties of convex faces in order to verify its efficiency. Since the mapping from the decision space to the objective space generally reduces the number of dimensions, ESAM generally performs better than the decision based search methods.

This paper presents empirical comparison of ESAM to an existing method [5] for solving MOLPs based on randomly generated problems. These results show that for small size problems ($p = 3, m = 3, n < 5$), Sayin's method has better CPU time even though that ESAM CPU time is very reasonable. However, as soon as the size of the problem increases ($p = 3, m = 3, n > 5$), ESAM outperforms the Sayins method. Moreover, Sayin's method does not generate efficient faces for problem larger than where ($p = 3, m = 10, n = 9$). We continued testing the ESAM method on larger MOLPs instances. It continues to generate efficient faces.

The proposed method could be improved by considering improvements to the pivoting simplex method, use of meta-heuristics and better computation algorithms (for example combining with solvers like CPlex or other powerful engines). We are considering examining other properties of efficient faces in order to reduce the computation complexity of the method.

References

[1] L. Li, K.K. Lai, A fuzzy approach to the multiobjective transportation problem, Comput. Oper. Res. 27 (1) (2000) 43–57.
[2] J. Dey, S. Mondal, M. Maiti, Two storage inventory problem with dynamic demand and interval valued lead-time over finite time horizon under inflation and time-value of money, Euro. J. Oper. Res. 185 (1) (2008) 170–194.
[3] P. Armand, Finding all maximal efficient faces in multiobjective linear programming, Math. Program. 61 (3) (1993) 357–375.
[4] P.L. Yu, M. Zeleny, The set of all non-dominated solutions in linear cases and a multicriteria simplex method, J. Math. Anal. Appl. 49 (2) (1975) 430–468.
[5] S. Sayin, An algorithm based on facial decomposition for finding the efficient set in multiple objective linear programming, Oper. Res. Lett. 19 (1996) 87–94.
[6] J.P. Dauer, R.J. Gallagher, A combined constraint-space, objective-space approach for determining high-dimensional maximal efficient faces of multiple objective linear programs, Euro. J. Oper. Res. 88 (1990) 368–381.
[7] H.P. Benson, Further analysis of an outcome set-based algorithms for multiple-objective linear programming, J. Optim. Theory Appl. 97 (1) (1998) 1–10.
[8] H.P. Benson, An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem, J. Global Optim. 13 (1998) 1–24.
[9] H. Yan, Q. Wei, J. Wang, Constructing efficient solutions structure of multiobjective linear programming, J. Math. Anal. Appl. 307 (2) (2005) 504–523.
[10] H.P. Benson, E. Sun, Pivoting in an outcome polyhedron, J. Global Optim. 16 (4) (2000) 301–323.
[11] V. Chankong, Y.Y. Haimes, Multiobjective Decision Making Theory and Methodology, Elsevier Science, New York, 1983.

[12] H.P. Benson, S. Sayin, Towards finding global representations of the efficient set in multiple objective mathematical programming, Naval Res. Logist. 44 (1997) 47–67.
[13] R.E. Steuer, Multiple Criteria Optimization: Theory, Computation and Application, John Wiley, New York, 1985.
[14] H. Isermann, The enumeration of the set of all efficient solutions for a linear multiple objective program, Oper. Res. Quarter. 28 (3ii) (1977) 711–725.
[15] T. Gal, A general method for determining the set of all efficient solutions to a linear vector maximum problem, Euro. J. Oper. Res. 1 (1977) 307–322.
[16] J. Philip, Vector maximization at a degenerate vertex, Math. Program. 13 (1977) 357–359.
[17] P. Armand, C. Malivert, Determination of the efficient set in multiobjective linear programming, J. Optim. Theory Appl. 70 (1991) 467–489.
[18] L. Pourkarimia, M.A. Yaghoobib, M. Mashinchi, Determining maximal efficient faces in multiobjective linear programming problem, J. Math. Anal. Appl. 354 (2009) 234–248.
[19] A.A. Foroughi, Y. Jafari, A modified method for constructing efficient solutions structure of molp, Appl. Math. Model. 3 (2009) 2403–2410.
[20] M. Ehrgott, J. Puerto, A.M. Rodriguez-Chia, Primal-dual simplex method for multiobjective linear programming, J. Optim. Theory Appl. 134 (3) (2007) 483–497.
[21] H.P. Benson, E. Sun, Outcome space partition of the weight set in multiobjective linear programming, J. Optim. Theory Appl. 105 (1) (2000) 17–36.
[22] H.P. Benson, Generating the efficient outcome set in multiple-objective linear programs: the bicriteria case, Acta Math. Vietnam. 22 (1997) 29–51.
[23] T. Gal, F. Geue, A new pivoting rule for solving various degeneracy problems, Oper. Res. Lett. 11 (1992) 23–32.